Integrating dynamic social networks and spatio-temporal models for risk assessment, wargaming and planning

Jacob Crossman and Robert Bechtel
Soar Technology
Ann Arbor, MI USA
[jcrossman, bob.bechtel]@soartech.com

Van Parunak and Sven Brueckner
Vector Research Center, a division of TTGSI
Ann Arbor, MI USA
[van.parunak, sven.brueckner]@newvectors.net

Abstract—**Most social network modeling tools are descriptive. That is, they focus on representing and visualizing entities and relationships given historic data. While such representations are valuable for understanding the structure of organizations and groups, they leave the process of projecting the entities' future actions within the context of the network entirely to the user. Building on technologies from DARPA's RAID and COMPOEX programs, we are working to fill this gap by developing a system to aid analysts in projecting future activities in space, time, and socio-political dimensions. Our approach uses a generative model of a social network, related physical processes, and the environment (a geo-spatial model). The nodes (entities) in our networks have goals, resources and locations. They act, using their resources, to achieve their goals. These models generate estimates of future behavior using faster than real time simulation within the network, process, and geo-spatial models. Using such a system, an analyst can more rapidly explore questions such as "What is the chain of relationships, processes and locations that sits between individual bomb parts and an IED attack?" and "If we crack down in a given region, where are new attacks likely to surface?" Our system does not replace analysts but helps analysts more rapidly and thoroughly explore potential futures, assess risk, and plan for contingencies.**

Keywords: Dynamic Social Networks, Social Networks, Multi-domain Modeling, Faster-than-realtime Simulation, Network Simulation, Future Projection, Wargaming, Analysis Aids, Agent-based Modeling, Polyagents

I. GAPS IN NETWORK ANALYSIS TOOLS

Social network analysis has grown to be a key tool in the battle against global terrorism, which is characterized by loosely, coupled networks of cells and regional groups. Among other advantages, social networks:

- Enable visualization of the structure complex, network oriented groups
- Facilitate understanding and exploitation of key nodes and relationships to defeat the network
- Identify new and hidden structures that could pose a future threat

To date the focus of social network science has been primarily on creating, visualizing, and identifying structural elements within these networks. Several tools, such as Analyst Notebook, have been built to aid analysts in exploring these structures. When used in this way, social networks become a tool for increasing situational awareness (SA), which is an important part of the analysis process. However, SA is only the first step in analysis processes as we show in Figure 1 (Figure 4-2 in [1] lays this out in greater detail). SA forms the input to further processes of projection (e.g. wargaming), planning, and collection that answer the following critical questions:

- What is going to happen next?
- What should I do about it?
- How do I know I'm right?

The answers to these questions form the core of actionable intelligence. As Figure 1 shows, these processes form a feedback loop that reduces uncertainty over time and allows decision makers to react quickly to changing events.

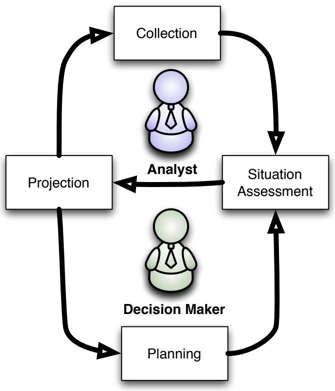


Figure 1: The analysis and decision making cycles.

Gaining SA is hard, requiring filtering and forensic analysis over large data sets. Later analysis steps are even harder. Projecting future activities of complex networks of humans is very nearly impossible at the finest levels of granularity (specific people, times, events) and is very difficult at higher levels (aggregate groups, wider time windows). Planning requires marrying this incomplete and partially accurate picture of what will happen with an assessment of risk against a hoped for beneficial payoff. Finally, these two processes must be integrated into a collection plan that ensures that those executing the plan are not blind to the changes and uncertainties inherent in the dynamics of the real world.

Despite these challenges, humans, and in particular intelligence analysts are asked to do just this on a daily basis. Typical questions an analyst may be asked include:

- How is the rise of leader A going to effect the IED activity in region B?
- Which would be more effective at reducing insurgent attacks in a region – an influence operation X or kinetic action against a resource Y?
- How is the population in region R likely to respond to course of action C?

With the exception of gaining better SA, analysts are left to answer these questions using their own processes and procedures. For example, to answer these questions an analyst may (among other steps) do the following:

- Review historical cases
- Hold a peer discussion and brainstorm
- Conduct an informal wargame
- Use situational logic to reason through alternatives

In general these processes used by analysts today are *ad hoc*, idiosyncratic, and time consuming. The best, most experienced analysts can do them well. But under time constraints, mountains of data, personal biases, and external pressures even the best analysts are unable to produce their best results. One possible outcome might be explored instead of three. An important second or third order effect might be ignored. Analysis may stop at the immediate effect and ignore the long-term effect, or it might look only at the local effect and ignore the regional effect. Even when done well, many of these tasks are carried out in an analyst's head without a reasoning trail or way to easily compare with another analyst's assessment. Biases creep in, assessments are watered down, and in general analysts are unable to keep up with the information load and operational tempo that characterize today's wars.

Tools and methods are needed to help analysts (and decision makers) with *all* of the steps in the analysis and planning cycle. In this paper we look at one such tool designed to help speed both the projection and collection planning processes with the goal of helping the analyst analysts answer critical questions more quickly, more thoroughly, more consistently, and with less bias.

II. Solution Characteristics

A complete solution requires tools that help an analyst project events and identify key indicators of those events. Several current technologies including Bayesian models, rule based systems, heuristic logic, and constructive simulation, can be used for these purposes, but there are currently few analyst end products that use these technologies. Each of these technologies has its advantages and can be applied effectively to different parts of the analyst's problems. In this paper we are agnostic with respect to these specific techniques and in fact use all of them in the system we will describe shortly.

However, before delving deeply into any specific solutions, it is worthwhile to examine the special characteristics of the analysis problem so that we have a basis on which to assess any solutions.

Special characteristics required for analysis include:

- *Ability to explore many futures rapidly*. Analysts need to consider several alternatives and then estimate the likelihood and risk of each; therefore projections processes must be *fast*.
- *Ability to automatically ingest and use data streams when available*. To the extent possible, any tool should leverage the existing streams of incoming data to automate its own configuration and setup as well as to learn the models required to run a projection.
- *Ability to trace reasoning*. Analyst conclusions are only as good as the assumptions underpinning those conclusions. It is very important to be able to trace conclusions back to the assumptions and data on which they were based.
- *Ability to measure the quality of the result*. Because projection, especially of human behavior, is a fundamentally uncertain activity, it is important to understand the information gaps and weaknesses in a projected future so the gaps can be filled and contingencies planned for.
- *Ability to recommend activities to reduce uncertainty*. If information gaps are measured, then it is possible to also suggest ways to fill those gaps through additional collection processes.
- *Ability to work across network, space, and time dimensions*. Though tools that work in only two of these dimensions can be useful, working in all three allows a system to be able to be more specific in its computations and recommendations.
- *Ability to customize models and change assumptions*. These types of tools are for generating and exploring hypotheses; therefore, they must allow variations to be constructed, projected, and compared.
- *Ability to work in both planning and real-time modes*. An ideal projection system would also tie to a fusion system that would automate the process of interpreting incoming data based on the hypotheses that have been built by the analyst

What makes the development of such a system most challenging is that all of these computational characteristics necessarily must be applied to *people*, their decisions, motivations, culture, and emotions. This has led some to believe that such systems are not possible to build or that if built, will be useless, or even worse, misleading. While this conclusion is understandable, we believe that it conflates two ways of using these types of systems. In this case we should not equate *predictive analysis* of well-understood physical systems where the computed output is used to *replace thinking* (e.g., as an oracle) with *anticipatory analysis* of highly complex systems where the output is used to *aid thinking* (e.g., as an advisory team). These types of systems have different goals and characteristics and thus different metrics must be used to evaluate each.

We like to compare this latter type of system, i.e. the aid to thinking, to a spreadsheet program used to project the next year's financial outlook. Since it is generally impossible to know exactly how customers, markets, personnel, and oversight organizations will behave, perfect prediction is highly unlikely. However, spreadsheets together with a few models, e.g., a conservative model, a most likely model, and a best/worst case model, can help a decision maker rapidly understand the range of possibilities, plan for contingencies, and reduce risk and the chance of surprise. Similarly, analysts equipped with appropriate tools to speed and facilitate thought processes can help prepare decision makers for alternative futures, interpret current trends, and assess and mitigate risk.

III. The DEFUSE System

We now present a candidate system for aiding analysts in projection and collection planning. Our goal for the system, called Defeating Enemy Forces United to Strike with Explosives (DEFUSE), is to provide analysts working in the IED domain (e.g. analysts at the Counter-IED Operations Integration Center (COIC)) with tools to rapidly assess future risks, explore contingencies, and plan efficient collection that can confirm or deny hypotheses about what might occur. DEFUSE, which is still being developed, is based on almost a decade of research and development building similar systems for both country level analysis and low-level tactical intelligence.

DEFUSE operates on a social network that is anchored in both time and space. That is, the nodes in the network, when appropriate, are associated with geographic locations as well as other attributes such as political and social affiliation. While DEFUSE provides interfaces for building networks directly and constructing initial networks automatically from data, the unique power of DEFUSE is its ability to rapidly simulate the behavior of a network over time. The outputs of the DEFUSE system are a time sequence of projected activities as well as a collection of specific named areas of interest (NAI) to search for indicating activities. The remainder of this paper will describe the DEFUSE architecture and foundational technologies, its capabilities and a simple scenario run using the system.

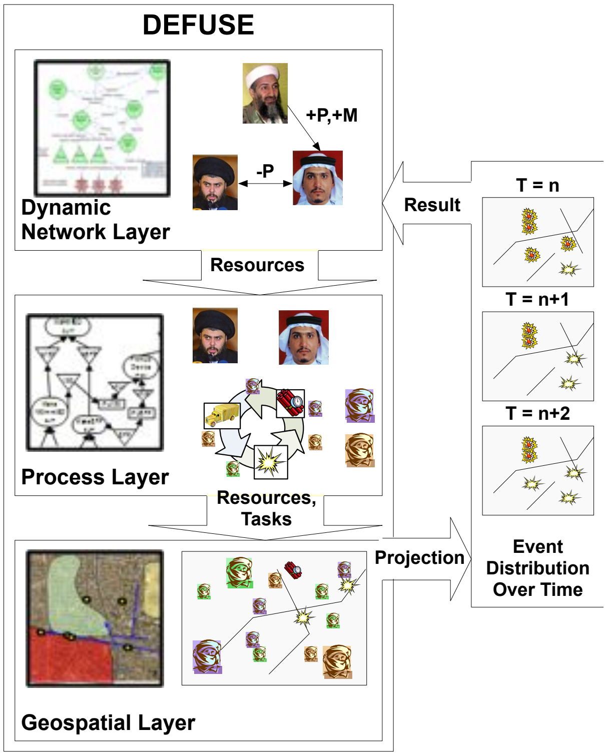


Figure 2: DEFUSE Architecture

A. The DEFUSE Architecture

DEFUSE's archiecture consists of three integrated layers (see also [2]). From top to bottom these layers are as follows:

- The dynamic network layer (DNL)
- The process layer (PL)
- The geospatial layer (GL)

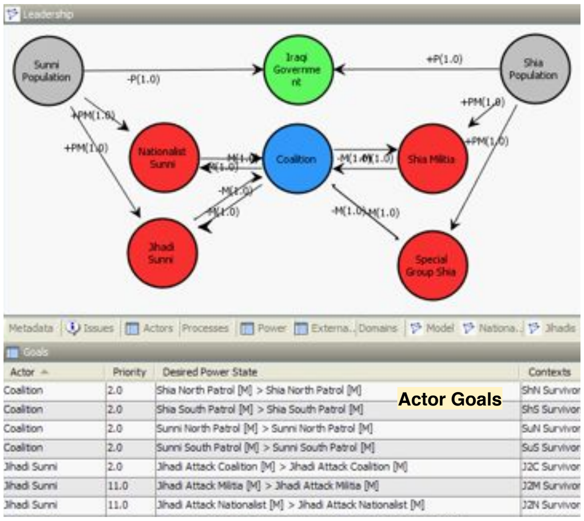
All three layers are faster than real-time simulations that can both represent the structure in their particular domain and rapidly project activities into the future. Each is based on existing technology that has been developed and tested independently of the others. The reason for using these three layers is that they constrain each other in ways that can help understand and project real world activities. The decision making logic and relationships at the DNL level provide traceable reasons for geo-spatial activities in the GL, while interactions within the GL limit what the DNL can compute in much the same way physical constraints put bounds on human actions.

Through simulation, the DEFUSE architecture is able to generate indicators up and down these layers of abstraction, leading to intelligence decision aids useful for tasking sensing assets or planning COAs. Next, we describe the technological structure and characteristics of each layer, referencing Figure 2 for illustration.

B. The Dynamic Network Layer

The dynamic network layer (DNL) consists of a network of entities called *actors* that represent leaders, key people, and populations. These networks can form any topological structure including hierarchies (i.e. top-level leaders and subordinates) and distributed cell networks. Connections between actors indicate relationships. These relationships can be friendly or hostile. In either case, the existence of a relationship within the model indicates some form of interaction may occur between those actors, for example transfer of orders, requests for aid, or declarations of threat. Given just these features, the leadership layer is a static social network. However, in DEFUSE each actor is also associated with a computational agent creating a dynamic network in which the nodes act to achieve their goals [3, 4]. The agent's medium of reasoning and exchange is power, where power can be divided among an arbitrary set of domains, such as political, military, economic, and social (PMES). Executing the model represented by the dynamic network generates an emergent outcome derived from agents seeking to optimize their own local success criteria as defined by their goals. This approach differs from traditional SNA algorithms, which apply top-down algorithms to meet global success criteria.

The basis for agents decision making is the computational theory of beliefs, desires, and intentions outlined in [5]. Each agent has a set of goals representing desired states in the world. The Al Qaeda agent representing an Al Qaeda cell might have a goal to have a greatly reduced US presence in Afghanistan. Each agent also holds a set of cause and effect beliefs, or rules about how activity in the world impacts the state of the world. For example, an Al Qaeda agent might have a belief that high casualty rates



Actor	Priority	Desired Power State	Contexts
Coalition	2.0	Shia North Patrol [M] > Shia North Patrol [M]	ShN Survivor
Coalition	2.0	Shia South Patrol [M] > Shia South Patrol [M]	ShS Survivor
Coalition	2.0	Sunni North Patrol [M] > Sunni North Patrol [M]	SuN Survivor
Coalition	2.0	Sunni South Patrol [M] > Sunni South Patrol [M]	SuS Survivor
Jihadi Sunni	2.0	Jihadi Attack Coalition [M] > Jihadi Attack Coalition [M]	J2C Survivor
Jihadi Sunni	11.0	Jihadi Attack Militia [M] > Jihadi Attack Militia [M]	J2M Survivor
Jihadi Sunni	11.0	Jihadi Attack Nationalist [M] > Jihadi Attack Nationalist [M]	J2N Survivor

Figure 4: Dynamic Network Layer

cause US withdrawal and that a high level of military aggressiveness (e.g. IED attacks) cause high casualty rates. Each agent then uses these goals and believes to select actions which increase the expected utility for that agent, where expected utility is defined as the extent to which an agent's goals are met. In our very simple example, an Al Qaeda agent would select to act to increase its military aggressiveness. By belief chaining, it believes this action will increase US casualties and thus cause the US to withdrawal. Thus agents take actions, which are logically consistent with their own beliefs and goals. However, because each agent can have its own beliefs and goals, these actions do not have to be logically consistent with any other agent's beliefs or goals (including US goals and beliefs).

One advantage of this approach is that it provides a computational mechanism for generating major shifts (or tipping points) because the agent decisions can be non-linear (e.g., an agent may dramatically shift its activities when a goal switches between met/unmet states). Furthermore, agent decisions are, in most cases, deterministic and traceable, making the decisions understandable to humans.

C. *The Process Layer*

The process layer (PL) forms the glue that ties the DNL and GL together. The process layer does two important things. First it provides a way to model physical process that are not necessary to model in spatial detail (e.g., building an IED) and integrate these processes with spatial processes. Second the PL models the resources needed to execute tasks. These resources are the primary connection point between the DNL and the GL as we will show.

PL models are hierarchical task networks (HTNs). The implementation basis for this network is an extension to the TÆMS graph [6] called gTÆMS, which is specifically designed to support coordination and interaction between groups of computational agents. Figure 3 shows an example gTÆMS network for a fraction of an IED creation and emplacement process. A gTÆMS graph is a bipartite graph; that is, it contains two separate classes of nodes and each node can only be connected to a node of the opposite type. gTÆMS supports task nodes (represented as ovals) and resource nodes (represented as triangles). Resources indicate

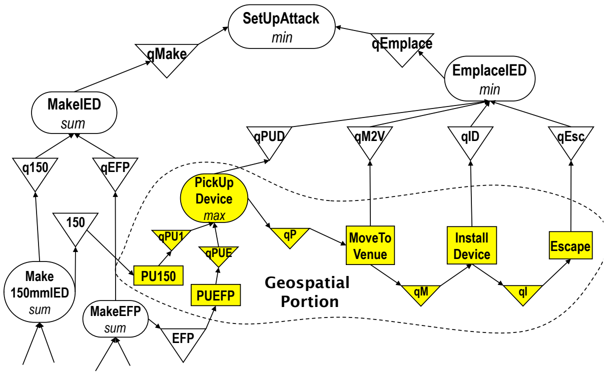


Figure 3: The Process Layer

constraints on activities, for example, a quantity of munitions that are available to be made into an IED. Tasks indicate activities that are either abstract (do not have a representation in our geo-spatial model) or concrete (having a representation in the geo-spatial model). For example, "Make 150mm IED" is abstract, as our geo-spatial model does not support simulation of the lowest level processes such as picking up two parts and putting them together. However, "Move to Venue" is concrete, as our environmental model does support moving from one point on a map to another. This distinction becomes important later when we discuss how agents move through this graph.

Two gTÆMS constructs make them useful for modeling coordinated behaviors. First, resources form a natural coordination point. That is, though agents may execute tasks in parallel, the resource nodes serve as points of coordination and competition and thus constrain agent activity to physically possible levels (e.g. given a specific quantity and type of resource). Second, the structure of the links between tasks and nodes forms a set of constraints on ordering and hierarchy. That is, there are some tasks that agents must do in sequence, while there are others they can do in parallel.

The process layer defines the structural bridge between strategic decision-making and IED team activity. DNL agents direct lower level activities by changing the resources available for a particular task. For example, a DNL agent might place the majority of its resources in the "EFP Parts" resource or might favor placing resources, in the form of additional people, on a branch of the graph that targets convoy activities. Spatially situated agents (discussed in detail below), move and act within the graph to execute various aspects of the IED process cycle such as bomb making and deployment.

D. *The Geospatial Layer*

The geospatial layer (GL) consists of a terrain database and a collection of "active terrain overlays." [7-9]. These overlays are scalar fields representing the distribution of a particular environment property in both space and time. For example, one field might represent the relative threat to an OPFOR entity. This threat has a spatial component represented as a distribution of a threat value over the map, as well as a temporal component, represented as a function controlling the change in that value over time. These fields form the virtual landscape over which the entities within the GL layer make decisions and move.

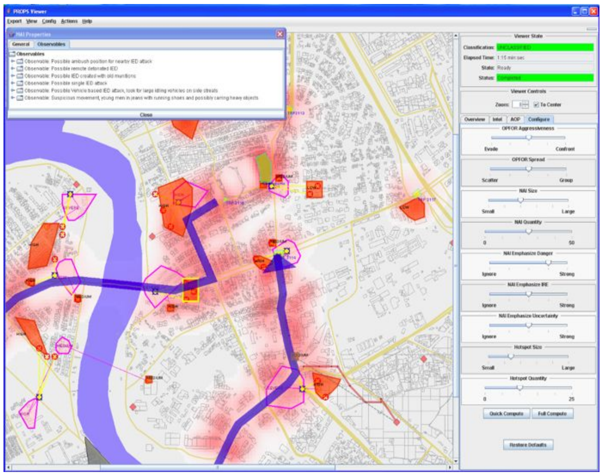
Entities representing individuals (e.g. bomb makers) and teams (e.g. deployment teams) are represented within DEFUSE as polyagents [10]. Polyagents have two components. First, they have an avatar representing an actual entity or team in the world. Second they can generate a set of ghosts, which represent that entity when projecting activities into the future. Each polyagent generates hundreds of ghosts in any single simulation run. Each ghost moves and acts in the world based on its personality – a simple set of scalar values that indicate the attractiveness or repulsion of a ghost to a given scalar field. Thus, the decision process for ghost agents is computationally very efficient (roughly equivalent to a dot product), and thousands of ghost entities can swarm and interact in real time. Furthermore, when real world data is available, these personality vectors can be automatically configured via genetic algorithms to generate the outcome distributions seen in real world data.

Ghosts operate in both the process and geospatial dimensions (Figure 5). The PL provides the coordination constraints, which guide the ghost's activities (e.g. telling the ghost what task comes next), while the GL model provides the physical space within which spatial activities occur.

After thousands of ghosts have acted within the GL, a probability distribution of activity is formed. This distribution can be aggregated in various ways to form useful conclusions about activities such as the likely attack locations or routes used for certain activities.

They can also be used to form named areas of interest (NAIs). By combining scalar fields representing projected opponent activity, uncertainty, and risk levels (e.g. from estimated impact on a course of action) the GL can construct and prioritize these NAIs for collection as shown by the orange polygons in *Figure 5*. Additionally, because the GL layer knows what activity was projected in a region, it can estimate observables for that region as well. These NAI computations provide a systematic and robust way of estimating the best search asset locations to best mitigate risk, reduce uncertainty, and lower the chance of surprise.

Thus, DEFUSE combines three powerful representational layers with two computational agent technologies to form a simulation framework capable of projecting a broad range of activities.



Observables

- Observable: Possible ambush position for nearby IED attack
- Observable: Possible remote detonated IED
- Observable: Possible IED created with old munitions
- Observable: Possible single IED attack
- Observable: Possible Vehicle based IED attack, look for large idling vehicles on side streets
- Observable: Suspicious movement, young men in jeans with running shoes and possibly caring heavy objects

Classification: UNCLASSIFIED
Elapsed Time: 1:15 min sec
State: Ready
Status: Completed

Figure 5: The Geospatial Layer

IV. Initial Capabilities and Insights

While the individual layers of the DEFUSE system, and in particular the dynamic network and geospatial layers, are robust and relatively mature, the combined system is still being developed. Our initial efforts have focused on working out the computational and architectural details of the cross layer interactions.

A. The Integration in Detail

The integrations of the different modeling layers is the core research problem in DEFUSE. How, for example, do you translate a high-level link between two actors or a power transfer into activity on the ground? How do you blend physical activities that do not require geospatial modeling with those that do and ensure that they stay in sync?

B. Process Link Nodes

Our answer for the first question is to define a new type of network node called the *process-link node*. A process-link node is shared between the DNL and PL layers. Within the DNL layer it is treated as an external process. Agents do not know exactly what happens in that node, but they do have a high-level representation of its effects on the rest of the network. Within the PL layer this node is a resource. As we have discussed above, resources constrain the execution of processes. As an example of how this works, consider how a network actor that could control the level of effort applied to a particular target. In this case we might construct a process-link node representing the number of people available for attacks on that target. By providing power (say military power) to this node, the PL can compute a proportional number of entities to use in simulating attacks.

Process-link nodes are also used to feed results of spatial simulation back to the network agents. The mapping is the same – in the PL the nodes are resources – but the actor in the network cannot directly control the power levels in this node. Instead changes to the level of resource in the PL cause proportional changes in power levels within the network. For example, a network node could learn about how successful its actions have been by observing "casualty" and "survivor" process-link node that are connected to the output of a combat process within the PL.

C. Geo-methods

Our answer to the second question of linking the PL methods to the GL activities is to define a special type of process node called a *geo-method*. A geo-method executes by placing ghosts that enter that method within a virtual geospatial environment and having them interact with other entities and events within the environment. For example, a "move to region" geo-method might move a set of entities as provided by the input resource node to a specific location on the map.

When executing these geo-methods thousands of potential futures are explored using the spatial layer's swarming technology and probability distributions are formed to represent those futures. At the end of the geo-method output resource levels are computed from these distributions and provided back to the PL.

D. Asynchronous Behavior

Because of the nature of the activities at each level, the different layers execute at different time scales. DNL activities are at roughly the granularity of days or weeks. The PL activities are at roughly the granularity of hours or days, and the GL activities may represent minutes or hours.

Full synchronous execution would cause the simulation to take too long to be useful, while completely asynchronous execution would drive the models out of synch. For example, the DNL would complete simulation of a month while the GL layer had only completed simulation of a single day. Our solution is to mix synchronous and asynchronous behavior as follows:

1. The DNL executes one cycle (one set of actions)
2. The PL and GL execute p *avatar* cycles, where p corresponds to the time step ratio between the PL/GL and the DNL cycles. In an avatar cycle an avatar agent executes a single method, selecting either to continue a previously executed method if it did not complete execution last cycle or starting a new method. If it is a geo-method the avatar executes it within the geospatial simulation. All avatars execute in parallel attempting to successfully complete their respective methods. Method completion is assessed in each cycle based on a *terminating condition*, e.g. reaching a geo-spatial destination.
3. For each avatar cycle, the PL and GL execute *g* ghost cycles, where g is a fixed value set to the number of ghost cycles required to drive reasonable avatar movement. A ghost cycle is one micro-step in the simulation and is very fast (roughly like a vector operation). In simple terms several ghost cycles are used to generate a probability distribution of where the actual avatar is likely to go next. For more information see [10].

Therefore we see that DEFUSE executes some aspects of the system synchronously (e.g. DNL and PL/GL) but others in parallel (e.g. the avatar cycles).

E. A Simple Illustrative Scenario

We have implemented a simple scenario set in Iraq based on the assessment in [11]. The purpose of this scenario was to test DEFUSE's ability to model a real world situation and to provide critical feedback on the implementation. Future work will focus on improving the system's breadth and operational utility.

The current system has several limitations as follows:

- It uses a simplified GL level to speed development. We will be swapping out the current GL for our more robust GL in future iterations.
- It does not generate NAIs (because it uses a simplified GL)

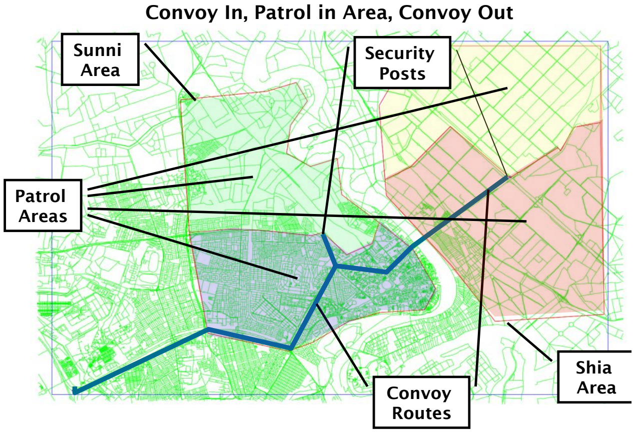


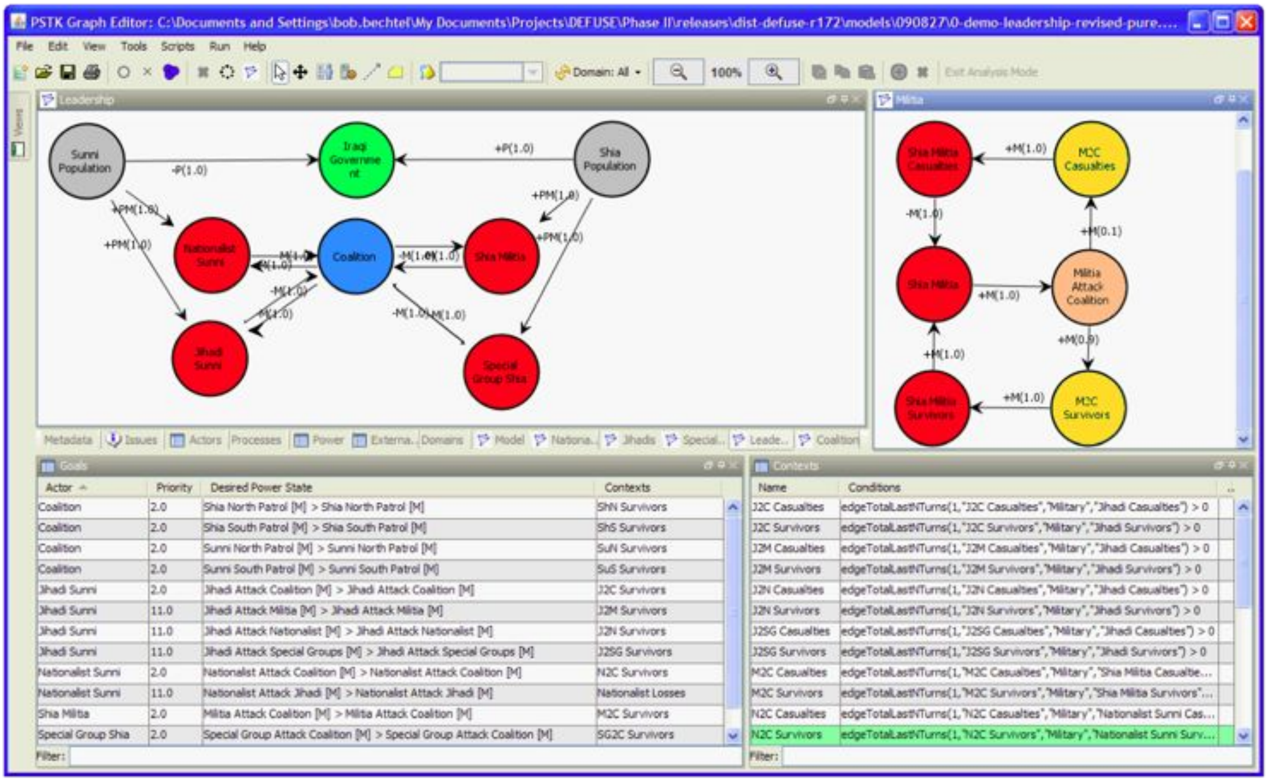
Figure 6: Scenario Region Overview

Our scenario is set in central Baghdad (Figure 6) where a brigade combat team (BCT) commander is responsible for maintaining security along a convoy route going to and from Baghdad International airport. The convoy route passes through a Sunni region (west side of the river) and a Shia region (east side of the river). The Sunni and Shia regions are populated by a mix of internal factions. The Sunni are divided between Nationalists and Jihadis. The Shia are split between a militia faction and special groups Shia who are highly skilled in irregular warfare and especially hostile to Coalition forces.

The Shia and Sunni groups are in general hostile toward each other (but at varying degrees depending on the subgroup). We set up the scenario to explore how IED activities on the ground might vary given that these hostilities were mild or severe.

We began our use of DEFUSE by building a network model of the situation. A retired region analyst supported our team in building this model (Figure 8). DEFUSE allows independent sub-views on the global network. Here we show two sub-networks as the entire network is too large to display clearly. On the left we see the general, high-level situation. Each of the factions is supported by its respective population with political and military support. The goals in the lower left pane are the primary driver of node actions, while the contexts, in the lower right pane, define when some of the goals are active.

The drill down view to the right of Figure 8 shows the use of the process-link nodes to integrate the network and process models. For example, the Militia Attack Coalition node is an input process-link node and binds to a resource node in the PL. The amount of military power transferred to this node directly influences the amount of human resources available to execute coalition attacks in lower layers. The yellow "M2C" nodes are output process-link nodes are bound to the results of the underlying processes. That is, these nodes are populated with the casualty levels derived from running the PL models. The edges between the nodes



Actor	Priority	Desired Power State	Contexts
Coalition	2.0	Shia North Patrol [M] > Shia North Patrol [M]	ShN Survivors
Coalition	2.0	Shia South Patrol [M] > Shia South Patrol [M]	ShS Survivors
Coalition	2.0	Sunni North Patrol [M] > Sunni North Patrol [M]	SuN Survivors
Coalition	2.0	Sunni South Patrol [M] > Sunni South Patrol [M]	SuS Survivors
Jihadi Sunni	2.0	Jihadi Attack Coalition [M] > Jihadi Attack Coalition [M]	J2C Survivors
Jihadi Sunni	11.0	Jihadi Attack Militia [M] > Jihadi Attack Militia [M]	J2M Survivors
Jihadi Sunni	11.0	Jihadi Attack Nationalist [M] > Jihadi Attack Nationalist [M]	J2N Survivors
Jihadi Sunni	11.0	Jihadi Attack Special Groups [M] > Jihadi Attack Special Groups [M]	J2SG Survivors
Nationalist Sunni	2.0	Nationalist Attack Coalition [M] > Nationalist Attack Coalition [M]	N2C Survivors
Nationalist Sunni	11.0	Nationalist Attack Jihadi [M] > Nationalist Attack Jihadi [M]	Nationalist Losses
Shia Militia	2.0	Militia Attack Coalition [M] > Militia Attack Coalition [M]	M2C Survivors
Special Group Shia	2.0	Special Group Attack Coalition [M] > Special Group Attack Coalition [M]	SG2C Survivors

Name	Conditions
J2C Casualties	edgeTotalLastNTurns(1,"J2C Casualties","Military","Jihadi Casualties") > 0
J2C Survivors	edgeTotalLastNTurns(1,"J2C Survivors","Military","Jihadi Survivors") > 0
J2M Casualties	edgeTotalLastNTurns(1,"J2M Casualties","Military","Jihadi Casualties") > 0
J2M Survivors	edgeTotalLastNTurns(1,"J2M Survivors","Military","Jihadi Survivors") > 0
J2N Casualties	edgeTotalLastNTurns(1,"J2N Casualties","Military","Jihadi Casualties") > 0
J2N Survivors	edgeTotalLastNTurns(1,"J2N Survivors","Military","Jihadi Survivors") > 0
J2SG Casualties	edgeTotalLastNTurns(1,"J2SG Casualties","Military","Jihadi Casualties") > 0
J2SG Survivors	edgeTotalLastNTurns(1,"J2SG Survivors","Military","Jihadi Survivors") > 0
M2C Casualties	edgeTotalLastNTurns(1,"M2C Casualties","Military","Shia Militia Casualtie...
M2C Survivors	edgeTotalLastNTurns(1,"M2C Survivors","Military","Shia Militia Survivors"...
N2C Casualties	edgeTotalLastNTurns(1,"N2C Casualties","Military","Nationalist Sunni Cas...
N2C Survivors	edgeTotalLastNTurns(1,"N2C Survivors","Military","Nationalist Sunni Surv...

Figure 7: A portion of the DNL model for the scenario

are used to estimate what will happen during agent decision-making. In this case the militia leader assumes that some of the expended military power will go toward lethal combat (upper square) and the rest will be expended in non-lethal activities. This model is only an approximation. The PL and GL do the actual attack simulation.

Drilling down into the PL we show one of the process models used to drive physical behavior (Figure 7). This model is a portion of the IED attack model. Here we see four processes (rectangles) and resources linking these processes. During execution, swarm agents move along the lines in the process model stopping at each process node to execute the given action. If the action is a spatial action (e.g. move), then the entity is placed on the map and executes the action, interacting with the terrain and other entities it finds along the way. The resource links serve to limit the activity within the process. One way to read this is that the processes are a function of the input resources. For example, the place IED process is a function of the IEDs available, the number of survivors, and the number of the own force casualties (the agents will not execute an attack if they suffer too many casualties). Notice also the line from IED explodes through Attack Enabled and back to Conduct IED Attack. This line enforces the constraint that agents must finish one IED process cycle before trying again.

We executed this model in two configurations. In the first configuration, the hostility between Iraqi factions was set as low by configuring the goals to be neutral toward each other. In the second we added goals to the Jihadi faction of the Sunni group to reduce the military power of the nationalist and Shia groups.

The results are shown in Figure 9 (a, b). In the first case we see that IED hotspots are identified in two patrol areas near the convoy routes as the factions are primarily focused on anti-coalition activities. In the second scenario, we see that the patrol focused IED threat is reduced (a lighter density in the display). Additionally we see new IED activity aimed at the Shia and Nationalist activity centers.

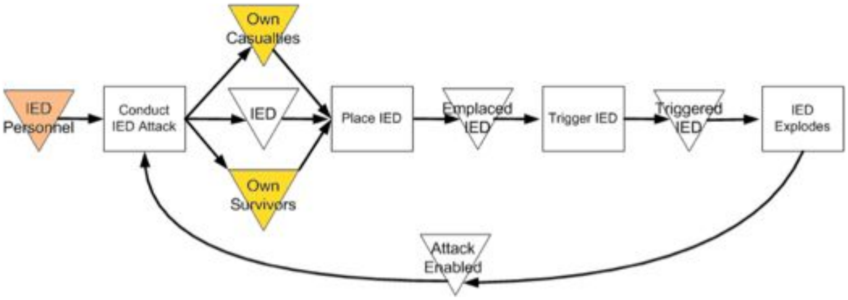
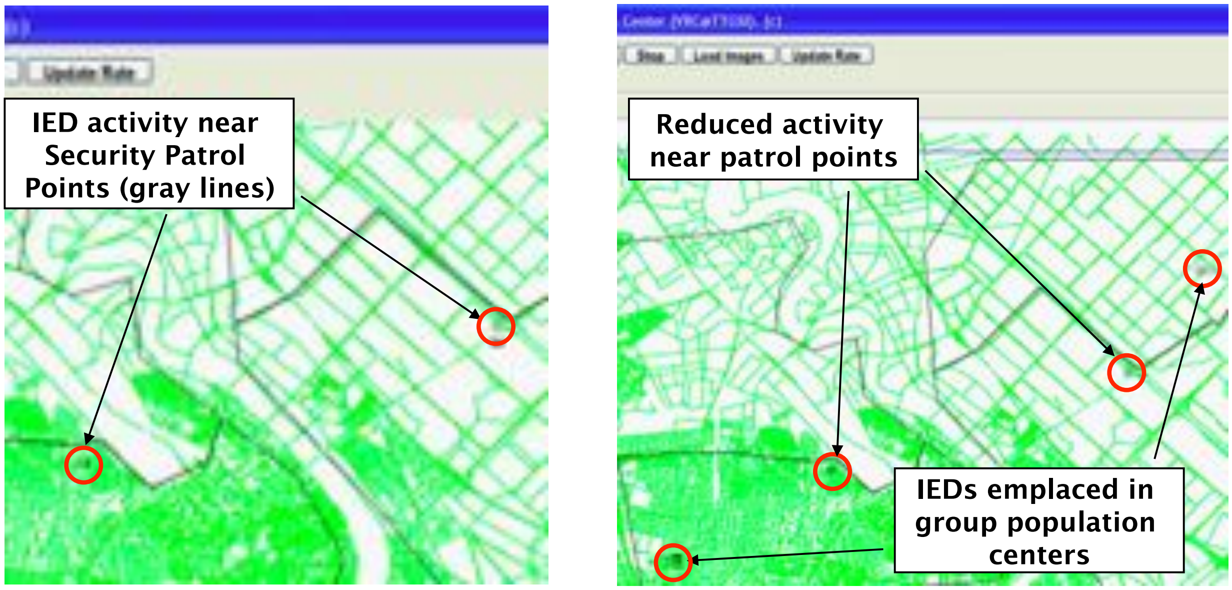


Figure 8: A portion of the PL model for the scenario


Update Rate
IED activity near Security Patrol Points (gray lines)
Stop
Load images
Update Rate
Reduced activity near patrol points
IEDs emplaced in group population centers

(a) *(b)*

Figure 9: Geospatial activity DEFUSE generated in simple scenario.

Though this outcome is not surprising, the outcomes can be generated rapidly and with a detail that an analyst cannot replicate by hand, and can track the effect of changes that a human cannot anticipate.

V. Looking Forward

With our initial development of the DEFUSE system, we are now poised to explore its capability in richer, more complex situations. Our near term goals include

- Implement larger scale scenarios and evaluations
- Integrate with our most capable and robust GL, enabling DEFUSE to generate NAIs.
- Increase DEFUSE's usability, by making DEFUSE models as easy as possible to construct [12], evaluate, and maintain.

Finally, we are also working to develop new ways to model human behavior that incorporate more of the non-kinetic features of human activities and decisions, for example, incorporating biased situation assessments and emotions into their behavior. As this research evolves, we will be incorporating the results into the DEFUSE line of products, providing a more comprehensive solution to analytic projection and risk assessment.

Acknowledgment

This work was sponsored by ONR (contract N00014-08-C-0774). Foundational efforts were sponsored by DARPA (contracts W15P7T-05-C-P032 and NBCH040153) and RDECOM-STTC (contract N61339-07-C-0016).

References

[1] Army, U., *Field Manual No 2-0, Intelligence*, D.o.t.A. Headquarters, Editor. 2004.

[2] Parunak, H.V.D., et. al, *Multi-Layer Simualtion for Analyzing IED Threats*, in *Proceedings of the IEEE Internatinoal Conference on Technologies for Homeland Security*. 2009: Waltham, MA.

[3] Taylor, G., et al. *Agent-based Simulation of Geo-Political Conflict*. in *16th Conference on Innovative Applications of Artificial Intelligence*. 2004. San Jose, CA: AAAI Press.

[4] Taylor, G. and E. Waltz. *A Framework for Modeling Social Power Structures*. in *14th Annual Conference for the North American Association for Computational Social and Organizational Sciences (NAACSOS)*. 2006. Notre Dame, IN.

[5] Rao, A.S. and M. Georgeff. *BDI Agents: From Theory to Practice*. in *The First International Conference on Multi-Agent Systems*. 1995. San Francisco.

[6] Nagendra, P., et al., *Exploring Organizational Designs with TAEMS: A Case Study fo Distributed Data Processing*, in *Second International Conference on Multi-agent Systems*. 1996: Kyoto, Japan.

[7] Parunak, H.V.D., et al. *Hybrid Multi-Agent Systems*. in *Proceedings of the Fourth International Workshop on Engineering Self-Organizing Systems (ESOA '06)*. 2006. Hakodate, Japan: Springer.

[8] Parunak, H.V.D. and S.A. Brueckner, *Extrapolation of the Opponent's Past Behaviors*, in *Adversarial Reasoning: Computational Approaches to Reading the Opponent's Mind*, A. Kott and W. McEneany, Editors. 2006, Chapman and Hall/CRC Press: Boca Raton, FL.

[9] Nielsen, P., et al., *Human Factors in Opponent's Intent*, in *Adversarial Reasoning: Computational Approaches to Reading the Opponent's Mind,*, A. Kott and W. McEneaney, Editors. 2006, CRC Press: Boca Raton, FL.

[10] Parunak, H.V.D. and S. Brueckner, *Concurrent Modeling of Alternatie Worlds with Polyagents*, in *Seventh International Workshop on Multi-Agent-Based Simulation (MABS06, at AAMAS06)*. 2006, Springer: Hakodate, Japan.

[11] Government, U., *Measuring Stability and Security in Iraq: Report to Congress in accordance with the Department of Defense Appropriations Act 2007 (Section 9010, Public Law 109-289)*, D.o. Defense, Editor. 2007, http://www.defenselink.mil/pubs/pdfs/9010_March_2007_Final_Signed.pdf.

[12] Taylor, G. and M. Quist, *Acquiring Agent-based Models of Conflict from Event Data*, in *IJCAI*. 2009, AAAI Press: Pasadena, CA.